

Input Program Syntax

December 2016

This document provides the grammar of the input language accepted by PRECiSA, which is a subset of the Prototype Verification System (PVS) language.

Lexical Structure

Literals

VarId literals are recognized by the regular expression $\langle upper \rangle (\langle letter \rangle | \langle digit \rangle | \text{'_'} | \text{'?'})^*$

NonVarId literals are recognized by the regular expression $\langle lower \rangle (\langle letter \rangle | \langle digit \rangle | \text{'_'} | \text{'?'} | \text{'@'})^*$

Reserved words and symbols

The set of reserved words is the set of terminals appearing in the grammar. Those reserved words that consist of non-letter characters are called symbols, and they are treated in a different way from those that are similar to identifiers. The lexer follows rules familiar from languages like Haskell, C, and Java, including longest match and spacing conventions.

The reserved words are the following:

AND	BEGIN	Dabs
Dadd	Darccos	Darcsin
Dcos	Ddiv	Dfloor
Dmod	Dmul	Dneg
Dsin	Dsqrt	Dsub
ELSE	END	ENDIF
FALSE	IF	IMPORTING
IN	LET	NOT
OR	PI	RtoD
RtoS	Sabs	Sadd
Sarccos	Sarcsin	Scos
Sdiv	Sfloor	Smod
Smul	Sneg	Ssin
Ssqrt	Ssub	THEN
THEORY	TRUE	VAR
abs	arccos	arcsin
cos	floor	mod
pi	sin	sqrt
unb_double	unb_nz_double	unb_nz_single
unb_pos_double	unb_pos_single	unb_single

The symbols are the following:

() +
 - * /
 ^ , =
 /= < <=
 > >= :

Comments

Single-line comments begin with %.

There are no multiple-line comments in the grammar.

Syntactic Structure

Non-terminals are enclosed between \langle and \rangle . The symbols $::=$ (production), $|$ (union) and ϵ (empty rule) belong to the BNF notation. All other symbols are terminals.

$$\begin{aligned} \langle AExpr \rangle & ::= \langle AExpr1 \rangle \\ & | \langle AExpr \rangle + \langle AExpr1 \rangle \\ & | \langle AExpr \rangle - \langle AExpr1 \rangle \end{aligned}$$

$$\begin{aligned}
\langle AExpr1 \rangle & ::= \langle AExpr2 \rangle \\
& | \langle AExpr1 \rangle * \langle AExpr2 \rangle \\
& | \langle AExpr1 \rangle / \langle AExpr2 \rangle \\
\langle AExpr2 \rangle & ::= \langle AExpr3 \rangle \\
& | \langle AExpr3 \rangle ^ \langle AExpr2 \rangle \\
\langle AExpr3 \rangle & ::= \langle AExpr4 \rangle \\
& | - \langle AExpr4 \rangle \\
\langle AExpr4 \rangle & ::= \langle AExpr5 \rangle \\
& | \text{floor} (\langle AExpr \rangle) \\
& | \text{sqrt} (\langle AExpr \rangle) \\
& | \text{abs} (\langle AExpr \rangle) \\
& | \text{sin} (\langle AExpr \rangle) \\
& | \text{cos} (\langle AExpr \rangle) \\
& | \text{arccos} (\langle AExpr \rangle) \\
& | \text{arcsin} (\langle AExpr \rangle) \\
& | \text{mod} (\langle AExpr \rangle , \langle AExpr \rangle) \\
\langle AExpr5 \rangle & ::= \langle AExpr6 \rangle \\
& | \text{pi} \\
& | \text{PI} \\
& | \langle Integer \rangle \\
& | \langle Double \rangle \\
\langle AExpr6 \rangle & ::= (\langle AExpr \rangle) \\
\langle ListFAExpr \rangle & ::= \langle FAExpr \rangle \\
& | \langle FAExpr \rangle , \langle ListFAExpr \rangle
\end{aligned}$$

```

⟨FAExpr⟩ ::= ⟨FAExpr1⟩
| Sadd ( ⟨FAExpr⟩ , ⟨FAExpr⟩ )
| Dadd ( ⟨FAExpr⟩ , ⟨FAExpr⟩ )
| Ssub ( ⟨FAExpr⟩ , ⟨FAExpr⟩ )
| Dsub ( ⟨FAExpr⟩ , ⟨FAExpr⟩ )
| Smul ( ⟨FAExpr⟩ , ⟨FAExpr⟩ )
| Dmul ( ⟨FAExpr⟩ , ⟨FAExpr⟩ )
| Sdiv ( ⟨FAExpr⟩ , ⟨FAExpr⟩ )
| Ddiv ( ⟨FAExpr⟩ , ⟨FAExpr⟩ )
| Sneg ( ⟨FAExpr⟩ )
| Dneg ( ⟨FAExpr⟩ )
| Sfloor ( ⟨FAExpr⟩ )
| Dfloor ( ⟨FAExpr⟩ )
| Ssqrt ( ⟨FAExpr⟩ )
| Dsqrt ( ⟨FAExpr⟩ )
| Sabs ( ⟨FAExpr⟩ )
| Dabs ( ⟨FAExpr⟩ )
| Ssin ( ⟨FAExpr⟩ )
| Dsin ( ⟨FAExpr⟩ )
| Scos ( ⟨FAExpr⟩ )
| Dcos ( ⟨FAExpr⟩ )
| Sarccos ( ⟨FAExpr⟩ )
| Darccos ( ⟨FAExpr⟩ )
| Sarcsin ( ⟨FAExpr⟩ )
| Darcsin ( ⟨FAExpr⟩ )
| Smod ( ⟨FAExpr⟩ , ⟨FAExpr⟩ )
| Dmod ( ⟨FAExpr⟩ , ⟨FAExpr⟩ )

⟨FAExpr1⟩ ::= ⟨FAExpr2⟩
| ⟨FAExpr3⟩ ^ ⟨FAExpr2⟩

⟨FAExpr2⟩ ::= ⟨FAExpr3⟩
| - ⟨FAExpr4⟩

⟨FAExpr3⟩ ::= ⟨FAExpr4⟩
| ⟨NonVarId⟩ ( ⟨ListFAExpr⟩ )
| ⟨NonVarId⟩

⟨FAExpr4⟩ ::= ⟨FAExpr5⟩
| RtoS ( ⟨AExpr⟩ )
| RtoD ( ⟨AExpr⟩ )

⟨FAExpr5⟩ ::= ⟨FAExpr6⟩
| ⟨VarId⟩

⟨FAExpr6⟩ ::= ( ⟨FAExpr⟩ )

```

$$\begin{aligned}
\langle BExpr \rangle & ::= \langle BExpr1 \rangle \\
& | \langle BExpr \rangle \text{ OR } \langle BExpr1 \rangle \\
\langle BExpr1 \rangle & ::= \langle BExpr2 \rangle \\
& | \langle BExpr1 \rangle \text{ AND } \langle BExpr2 \rangle \\
\langle BExpr2 \rangle & ::= \langle BExpr3 \rangle \\
& | \text{ NOT } \langle BExpr3 \rangle \\
\langle BExpr3 \rangle & ::= \langle BExpr4 \rangle \\
& | \langle FAExpr \rangle = \langle FAExpr \rangle \\
& | \langle FAExpr \rangle \neq \langle FAExpr \rangle \\
& | \langle FAExpr \rangle < \langle FAExpr \rangle \\
& | \langle FAExpr \rangle \leq \langle FAExpr \rangle \\
& | \langle FAExpr \rangle > \langle FAExpr \rangle \\
& | \langle FAExpr \rangle \geq \langle FAExpr \rangle \\
\langle BExpr4 \rangle & ::= \langle BExpr5 \rangle \\
& | \text{ TRUE} \\
& | \text{ FALSE} \\
\langle BExpr5 \rangle & ::= (\langle BExpr \rangle) \\
\langle FPtype \rangle & ::= \text{ unb_single} \\
& | \text{ unb_double} \\
& | \text{ unb_pos_single} \\
& | \text{ unb_pos_double} \\
& | \text{ unb_nz_single} \\
& | \text{ unb_nz_double} \\
\langle ListArg \rangle & ::= \langle Arg \rangle \\
& | \langle Arg \rangle , \langle ListArg \rangle \\
\langle Arg \rangle & ::= \langle ListVarId \rangle : \langle FPtype \rangle \\
& | \langle VarId \rangle \\
\langle Args \rangle & ::= \langle ListArg \rangle \\
\langle ListVarId \rangle & ::= \langle VarId \rangle \\
& | \langle VarId \rangle , \langle ListVarId \rangle \\
\langle Stm \rangle & ::= \langle Stm1 \rangle \\
& | \text{ LET } \langle VarId \rangle = \langle FAExpr \rangle \text{ IN } \langle Stm \rangle \\
\langle Stm1 \rangle & ::= \langle Stm2 \rangle \\
& | \text{ IF } \langle BExpr \rangle \text{ THEN } \langle Stm \rangle \text{ ELSE } \langle Stm \rangle \text{ ENDIF} \\
\langle Stm2 \rangle & ::= \langle Stm3 \rangle \\
& | \langle FAExpr \rangle
\end{aligned}$$

$\langle \text{Stm3} \rangle ::= (\langle \text{Stm} \rangle)$

$\langle \text{ListDecl} \rangle ::= \langle \text{Decl} \rangle$
| $\langle \text{Decl} \rangle \langle \text{ListDecl} \rangle$

$\langle \text{Decl} \rangle ::= \langle \text{NonVarId} \rangle (\langle \text{Args} \rangle) : \langle \text{FPtype} \rangle = \langle \text{Stm} \rangle$
| $\langle \text{NonVarId} \rangle : \langle \text{FPtype} \rangle = \langle \text{Stm} \rangle$

$\langle \text{ListNonVarId} \rangle ::= \langle \text{NonVarId} \rangle$
| $\langle \text{NonVarId} \rangle , \langle \text{ListNonVarId} \rangle$

$\langle \text{Imp} \rangle ::= \text{IMPORTING } \langle \text{ListNonVarId} \rangle$

$\langle \text{VarDecl} \rangle ::= \langle \text{VarId} \rangle : \text{VAR } \langle \text{FPtype} \rangle$

$\langle \text{ListVarDecl} \rangle ::= \epsilon$
| $\langle \text{VarDecl} \rangle \langle \text{ListVarDecl} \rangle$

$\langle \text{Program} \rangle ::= \langle \text{NonVarId} \rangle : \text{THEORY BEGIN } \langle \text{Imp} \rangle \langle \text{ListVarDecl} \rangle \langle \text{ListDecl} \rangle \text{ END } \langle \text{NonVarId} \rangle$
| $\langle \text{NonVarId} \rangle : \text{THEORY BEGIN } \langle \text{ListVarDecl} \rangle \langle \text{ListDecl} \rangle \text{ END } \langle \text{NonVarId} \rangle$